PROJECTIVE METHODS TO FIND A FEASIBLE SOLUTION TO SYSTEMS OF LINEAR CONSTRAINTS

G. van der HOEK and J. TELGEN

REPORT 7620/0

ERASMUS UNIVERSITY ROTTERDAM, P.O. BOX 1738, ROTTERDAM, THE NETHERLANDS
REPORT 7620/0

PROJECTIVE METHODS TO FIND A FEASIBLE SOLUTION
TO SYSTEMS OF LINEAR CONSTRAINTS

by Gerard van der Hoek and Jan Telgen
PROJECTIVE METHODS TO FIND A FEASIBLE SOLUTION TO SYSTEMS OF LINEAR CONSTRAINTS

by

Gerard van der Hoek and Jan Telgen

paper to be presented at EURO II, Stockholm 1976

Contents

1. Summary .......................... 1
2. Introduction ....................... 1
3. The method of orthogonal projection 2
4. The reflexion method ............... 4
5. The FALCON algorithm .............. 5
6. Some preceding considerations .... 7
7. Computational tests ............... 9
8. Results .................................. 10
9. Concluding remarks ............... 12
References ........................... 13
1. Summary.

The problem of finding a feasible solution to systems of linear constraints is often encountered in operations research. In 1954 S. Agmon [1], T.S. Motzkin and I.J. Schoenberg [2] published articles about "the relaxation method for linear inequalities". These articles are often referenced. They propose an iterative method to find a feasible solution for feasible systems of linear inequalities. The authors prove that their methods either find a feasible point in a finite number of steps or converge to a feasible point (if one exists). Independently, we have developed another method [3], which uses orthogonal projection on violated constraints. However, a restriction, once fulfilled, will never be violated by following iteration points. This "FALCON algorithm" is proved to terminate after a finite number of steps, both for feasible and infeasible systems.

We intend to discuss here the various (dis-)similarities of these methods and to compare their theoretical background and computational efficiency.

2. Introduction.

The problem of finding a feasible solution to a system of linear constraints is well known and often encountered in mathematical programming. For example in linear programming it is a basic problem both in developing a starting solution for an iterative procedure towards the optimum and in identifying redundant constraints [4], [5]. Apart from that these problems also rise in analysis and engineering. This problem can be formulated as:

\[(1) \quad \text{find a vector } x^* \text{ such that } \sum_{j=1}^{n} a_{ij} x^*_j < b_i \text{ for all } i \in \{1, \ldots, m\}.
\]

It should be noted that neither \( x^* \) nor \( b \) is restricted to be positive in this formulation. Positivity restrictions should be explicitly mentioned in these restrictions. All systems of linear constraints can be formulated in such a way.
In addition to (1) we define the hyperplane $\pi_i$ as the set of points that satisfy the $i$-th constraint as an equality:

\[
\pi_i = \{ \mathbf{x} \mid \sum_{j=1}^{n} a_{ij} x_j = b_i \} \\
\text{for } i \in \{1, \ldots, m\}.
\]

In 1954 S. Agmon [1] and T.S. Motzkin and I.J. Schoenberg [2] published two articles about relaxation methods to find a feasible solution for the problem (1), using orthogonal projection on some hyperplanes (2). Independently the authors developed the so called FALCON algorithm [3], which is based on some similar ideas, but follows a different line of attack and obtains more complete results.

After a description of these methods we want to discuss their various (dis-)similarities and to compare their computational efficiency in some randomly generated problems.

3. The method of orthogonal projection.

S. Agmon defined his method of orthogonal projection [1] to solve (1) in the following way:

step 1: choose the starting point $\mathbf{x}^0$ arbitrarily;
step 2: check $\mathbf{x}^k$ on violation of the constraints of (1);
   if $\mathbf{x}^k$ is a solution to (1) the algorithm terminates with $\mathbf{x}^* = \mathbf{x}^k$,
   otherwise:
step 3: select the hyperplane $\pi_{i_0}$, related to the restriction $i_0$, with respect to which $\mathbf{x}^k$ has the highest degree of infeasibility;
step 4: let $\mathbf{q}^k$ be the orthogonal projection of $\mathbf{x}^k$ on $\pi_{i_0}$ that is

\[
q^k_j = x^k_j + (b_0 - \sum_{j=1}^{n} a_{i_0 j} x^k_j) \frac{a_{i_0 j}}{\sum_{j=1}^{m} (a_{i_0 j})^2}
\]

\[
\forall j \in \{1, \ldots, m\}
\]
then

\[(4) \ x^{k+1} = x^k + \lambda(a^k - x^k)\]

where \(0 < \lambda < 2.\)

and go to step 2.

In the method of orthogonal projection \(\lambda\) is usually taken to be equal to 1. Agmon proves this method to produce a (possibly infinite) sequence of points \(\{x^0, x^1, x^2, \ldots\}\), either terminating after a finite number of steps or converging to a solution to (1). (In both cases under the assumption that a solution to (1) exists.)

One important fact should be noted here: in the proof of the convergence no reference at all is made to the choice of the hyperplane related to the constraint with the highest degree of infeasibility. So this choice is of no influence on the convergence and one can arbitrarily choose any other hyperplane in step 3, with respect to which \(x^k\) is on the wrong side.

The method of orthogonal projection is illustrated by the following two figures.

The shaded region is the set of feasible points \(\{x^*\}\).
Starting from \(x^0\) (step 1) one first projects on \(\pi_1\) (step 3) resulting in \(x^1\) (step 4), and then on \(\pi_2\) (step 3) resulting in \(x^2\) (step 4). As \(x^2\) is feasible: \(x^2 = x^*\) (step 2).
The method terminates in a finite number of steps.

The shaded region is the set of feasible points \(\{x^*\}\).
Starting from \(x^0\) (step 1) one first projects on \(\pi_2\) (step 3) resulting in \(x^1\) (step 4), then on \(\pi_1\) (step 3) giving \(x^2\) (step 4), then again on \(\pi_2\) (step 3) etc.
This (infinite) sequence \(\{x^0, x^1, x^2, \ldots\}\) converges to a point \(x^*\) of the feasible region.
4. The reflexion method.

The reflexion method [2] of T.S. Motzkin and I.J. Schoenberg closely resembles the method of orthogonal projection. However there is one interesting and important difference. In step 4 of the method one does not project $x_k$ on a hyperplane $\pi_{i_0}$ corresponding to a violated constraint $i_0$, but one reflexes $x_k$ with respect to this hyperplane $\pi_{i_0}$, to obtain $x_{k+1}$.

In fact this is the same as putting $\lambda = 2$ in step 4 of the method of orthogonal projection; so that the formula:

$$x_{k+1}^j = x_k^j + 2(b_{i_0} - \sum_{l=1}^{n} a_{i_0 l} x_l^k) \frac{a_{i_0 j}}{\sum_{l=1}^{n} (a_{i_0 l})^2}$$

for all $j \in \{1, ..., n\}$

can be used instead of (3) and (4).

The restriction on the value of $\lambda$ in this method leads to more exact results concerning the finiteness and the convergence of the sequence of points \{$x^0, x^1, x^2, ...$\} generated by this method.

For feasible systems Motzkin and Schoenberg prove that the reflexion method generates a sequence of points that terminates in a feasible solution in a finite number of steps, unless the dimension of the feasible region is smaller than the dimension of the solution space. In the latter case the sequence either terminates in a feasible solution after a finite number of steps or starts oscillating between two points that are symmetric with respect to the feasible region. (Note that at least one feasible solution is assumed to exist!) For this method, the same observation can be made as for the method of orthogonal projection: the selection of the hyperplane $\pi_{i_0}$ is not restricted to the one corresponding to the restriction $i_0$ with $i_0$ the highest degree of infeasibility by the way in which finiteness and convergence are proved. So one can alter the selection criterion for $\pi_{i_0}$ in step 3, as long as a hyperplane corresponding to a constraint violated by the current iteration point $x_k$ is chosen.

The following two figures illustrate the method:
The shaded area is the feasible region \( \{x^*\} \). Starting from \( x^0 \) (step 1) one reflexes on \( \pi_1 \) (step 3) resulting in \( x^1 \) (step 4), then reflexion on \( \pi_2 \) (step 3) results in \( x^2 \), etc, until one reaches \( x^5 = x^* \) (step 2).

The feasible region is the line \( \pi_1 \) (of lower dimension than the solution space) so one oscillates between \( x^0 \) and \( x^1 \).

---

5. The FALCON algorithm.

The FALCON algorithm is a more elaborate method to find a solution to the problem (1). Rather than regarding one constraint at a time, FALCON deals with more constraints simultaneously if necessary. The FALCON algorithm is also a projection method as will be clear from the following description:

- \( m \) is the number of restrictions
- \( k \) is an index counter

\begin{itemize}
  \item step 1: choose \( x^0 \) arbitrarily: \( k = 0 \)
  \item step 2: if \( k = m \), then the algorithm terminates \( x^k \) being a feasible solution of (1) otherwise set \( k = k + 1 \).
  \item step 3: check if \( x^{k-1} \) violates constraint \( k \); if \( x^{k-1} \) fulfils constraint \( k \), then \( x^k = x^{k-1} \); go to step 2 otherwise
  \item step 4: project \( x^{k-1} \) on the hyperplane \( \pi_k \) using formula (3), yielding \( a^1 \)
  \item step 5: if \( a^1 \) satisfies the first \( k \) constraints then \( x^k = a^1 \); go to step 2, otherwise
\end{itemize}
step 6: a finite search procedure is started in the intersection of the k-th
constraint and those previously added constraints, that are violated by
\( \mathbf{q}^{1} \). This finite search procedure uses the same concepts as the foregoing
steps: define \( C_1 = \pi^1 \). Then it projects \( x^{k-1} \) on \( C_2 = C_1 \cap \{ \text{hyperplane}
\}


\begin{equation}
q^{k+1}_j = x^{k-1}_j + \sum_{r \in C_\mathbf{q}} (b_r - \sum_{t=1}^{n} a_{rt} x^{k-1}_t) a_{sj} \sum_{s \in C_\mathbf{q}} r \in C_\mathbf{q}
\end{equation}


\begin{equation}
P^\mathbf{q} = \left\{ p_{rs} \right\} = \left\{ \sum_{j=1}^{n} a_{sj} a_{sj} \right\} \quad s, r \in C_\mathbf{q}
\end{equation}


As the dimension associated with \( C_\mathbf{q} \) is strictly decreasing in \( \mathbf{q} \), this
search procedure is finite and can also determine infeasibility.

In this way one always adds a constraint to the set of satisfied constraints
or determines infeasibility. So in a finite number of steps either a feasible \( x^* \)
is obtained or infeasibility is reported.

Although FALCON seems to be rather complicated the computations are simplified,
for one does not need to specify the intersection \( C_\mathbf{q} \) explicitly, as we are only
interested in the coordinates of \( \mathbf{q}^{k-1} \). Then the only difficulty is the matrix \( Q^\mathbf{q} \), but
this matrix is symmetric as can easily be seen from (7) and moreover there is a
great deal of familiarity between the matrix \( Q^\mathbf{q} \) and the matrix \( P^{\mathbf{q}+1} \), because \( P^{\mathbf{q}+1} \)
can be obtained from \( P^\mathbf{q} \) by adding one column and one row (which are identical).
These facts enable us to use a simple updating device for the inverse of symmetric
matrices (e.g. Frisch [6], Fletcher [7]), so the inversion routine does not have
to be gone through each time. A graphical example is shown in figure 5.
The shaded region is feasible. Starting from $\mathbf{x}^0$ (step 1) projection of $\mathbf{x}^0$ on $\pi_1$ gives $\mathbf{a}^1$ (step 4) which is $\mathbf{x}^1$ (step 5).

Projection of $\mathbf{x}^1$ on $\pi_2$ gives $\mathbf{a}^1$ (step 4) but this violates restriction 1 (step 5).

$\mathbf{a}^2$ is found in the intersection of $\pi_1$ and $\pi_2$ (step 6).

We see that $\mathbf{x}^2 = \mathbf{a}^2$ is feasible.

6. Some preceding considerations.

Before we turn to the computational comparison of the proposed methods some remarks should be made:

(I) As regards the complexity of the arithmetic to be performed it is clear that the method of orthogonal projection and the reflexion method are to be preferred above the FALCON algorithm.

(II) Next a very important matter should be mentioned. The method of orthogonal projection and the reflexion method cannot handle inconsistent or infeasible systems. In practice whether a problem is feasible or not cannot be deducted with these methods. If one does not find a feasible solution the problem may be infeasible, but the method may also produce a sequence $\{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots\}$ with one of the 3 following properties:

(1) it will terminate after a finite number of steps where finite is "infinite" for practical purposes;

(2) it is still converging;

(3) it will oscillate after a finite number of steps, where finite can also mean "infinite" for practical purposes.

So before applying these methods one first has to know whether the problem is feasible or not. But feasibility is usually proved by finding a feasible solution so there is no sense in using these methods if one already knows a feasible point, while there is no point at all in applying these methods to infeasible problems. On the other hand FALCON finds out whether a system is feasible or not and if it is feasible FALCON provides a feasible solution.

(III) As is also mentioned above, the three projection methods do not always produce the same conclusion. The most important difference is the uncertainty in the orthogonal projection method and the reflexion method. If one cannot find
a feasible solution, this does not always mean that there is no feasible solution. There is a certain amount of uncertainty in the "I can't find a feasible solution"-answer of these methods. On the other hand FALCON always yields a correct unique answer to the problem (1), whether it is feasible or not.

(IV) One more difference is the way in which the methods deal with the constraints. If in the FALCON algorithm a restriction is fulfilled by an iteration point $x^k$ then it will be fulfilled by all succeeding iteration points $x^\ell$ ($\ell > k$). In the other methods the satisfied constraints are not "guarded" against violation.

(V) Another major difference between these three methods lies in the way in which they handle equality constraints. There are two possibilities:

a) To treat equality constraints as two inequality constraints:

In the orthogonal projection method and the reflexion method this seems only possible when one projects on the constraint, so $\lambda = 1$, otherwise it is practically impossible to fulfil this constraint. However even when $\lambda = 1$ there may rise (very) great difficulties, for example when there are two equality constraints in two-dimensional space

![Diagram](image)

In the FALCON algorithm, equality constraints are handled in quite the same way as inequality constraints.

b) One can also use equality restrictions to eliminate some variable. This eliminated variable need not be unrestricted (like in most LP codes), because all constraints including positivity restrictions have to be stated explicitly in the relaxation methods. So a substitution of equality constraints leads to an actual reduction of the size of the remaining problem. This should compensate the extra calculations (comparable to one iteration in l.p.) which have to be performed in removing the equality constraints.

(VI) The independently developed FALCON algorithm can be seen as an extension of the orthogonal projection method insofar as it combines the profitable features of
this method with the exhaustiveness of the finite search procedures and therefore
gives a unique answer. Only a computational comparison can show the effectiveness
of this combination.

7. Computational tests

Computational tests have been performed on an IBM 1130 and an IBM 370/158. All
methods were programmed in Fortran IV and problems were generated randomly,
with density of non-zero coefficients varying from 10 to 80 per cent. We solved
some 1000 problems of size 29x18 and 12 problems of size 180 x 100 (more rows
than columns since the positivity constraints should be written down explicitly). Further
tests on larger problems are being performed.

In all projective methods mentioned before, there are a number of possible
variations (e.g., the λ's, the order in which the constraints are considered etc). So
first we tried to determine the best variant for each method and after that to
compare these variants. This led to the following results.

(I) In the orthogonal projection method and the reflexion method there is no
apparent good reason to project on the most violated constraint, so that is why
we also formulated a different variant that projects the current iteration point
on the next violated constraint.

(II) Furthermore a range of different λ's varying from 1. to 2. were used to
examine the behaviour of these methods. It will be clear that λ < 1 is of no
practical interest at all!

(III) We also incorporated a technique introduced in 1957 by R. Frisch [6] to
obtain a good starting point by a single projection. This technique, which we called
P-START, exists in projecting an arbitrarily chosen starting point \( \tilde{x} \) on the
intersection of all constraints violated by \( \tilde{x} \) to obtain a better starting point \( \tilde{x}^0 \).

(IV) With regard to the FALCON algorithm, we also programmed some variants. First
we considered it with and without P-START; next we investigated the influence of
the order in which the constraints are considered in the iteration process. We
compared the original order with the ordering of the constraints in decreasing
infeasibility.
8. Results.

As could be expected both in the method of orthogonal projection and the reflection method the more elaborate versions (including P-START, and projecting on the most violated constraint) were superior to the other variants with respect to the number of projections before a feasible point was reached. We obtained the following ordering of the variants regarding the number of projections: (with an increasing number of projections)

1. use P-START; project on the most violated constraint
2. use P-START; project on the next violated constraint
3. project on the most violated constraint
4. project on the next violated constraint.

However when we look at the time taken by these variants, it is clear that searching for the most violated constraint is a very time consuming affair. Besides that, application of P-START also takes some time, comparable with about 20 iterations. Regarding the time consumed by the variants we obtained the following ordering (increasing time consumed):

1. project on the next violated constraint
2. use P-START; project on the next violated constraint
3. use P-START; project on the most violated constraint
4. project on the most violated constraint.

As the time taken by these methods usually is a more important criterion than the number of projections, we think that one should prefer the simpler rule of projecting on the next violated constraint, above the rules suggested in [1], [2].

Within this variant there is a possibility of varying the $\lambda$. We think the variant with $0 \leq \lambda < 1$ to be of no practical meaning, so we only considered the case where $1 \leq \lambda \leq 2$. There is a very interesting result of our experiments with this variant (that applies for the other variants as well) and this can best be shown by a figure.
The scaling of the vertical axis of this figure depends on the problem under consideration (some typical values at $\lambda = 1$ and $\lambda = 2$ were 30 and 5 for our test problems). In almost all of our hundreds of test problems we found the relationship between $\lambda$ and the number of projections to be as in figure 7. From these tests it is clear to us that $\lambda$ should be chosen close to 2. So one has to "overproject" a lot to reach a feasible solution sooner. Perhaps this result can be of use in phase I of linear programming routines. Regarding the FALCON algorithm, we found from our tests that the use of P-START frequently leads to smaller solution times. The use of the device for re-ordering the constraints in a sequence of decreasing infeasibility, usually enlarges the solution time just as happened in the other projective methods.

In comparing the best variant of the orthogonal projection method and the reflexion method (the reflexion method with projection on the next violated constraint), and the best variant of the FALCON algorithm (FALCON including P-START) there are two important aspects to be noted. First, we only considered feasible problems, but still in about 25% of the problems the reflexion method could not find a feasible point in 1000 iterations. Second, in 15% of the cases the FALCON algorithm was faster than the reflexion method, so in the remaining 60% the reflexion method was the better one.
9. **Concluding remarks.**

If one has to determine whether a problem in linear restrictions is feasible or not and one wants to use a projective method, then there are several considerations.

On the one hand in 60% of the cases, where the system is feasible the method with reflexion on the next violated constraint discovers this fact sooner than the FALCON algorithm.

On the other hand the FALCON algorithm always yields a correct answer, even if the system is infeasible. This last aspect leads us to conclude that it is advisable to use the FALCON algorithm, because the difference in the solution time is not extreme and FALCON is exhaustive.

There are several proposals (e.g. [8]) to speed up the convergence of the orthogonal projection methods, in order to work away the disadvantages of these methods. These are mostly heuristic rules, that can be of use in special problems, but are too problem-oriented to be incorporated here.

Apart from the FALCON algorithm, we don't know a projective method that completely avoids the very slow convergence and inability to deal with inconsistent systems.

For small problems (up to $10 \times 20$) FALCON even outperforms the normal linear programming phase I procedure, but for larger problems methods like big M are still to be preferred.
References.


<table>
<thead>
<tr>
<th>No.</th>
<th>Code</th>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7600</td>
<td></td>
<td>List of Reprints; nos. 172-178, List of Reports; 1975.</td>
<td></td>
</tr>
<tr>
<td>7601</td>
<td>ES</td>
<td>&quot;Profit Structure in Retailing&quot;, by B. Nooteboom.</td>
<td></td>
</tr>
<tr>
<td>7602</td>
<td>M</td>
<td>&quot;On a Diophantine Equation of E. Bombieri&quot;, by R.J. Stroeker.</td>
<td></td>
</tr>
<tr>
<td>7603</td>
<td>S</td>
<td>&quot;Limit Theory for Multivariate Sample Extremes&quot;, by L. de Haan and S.I. Resnick.</td>
<td></td>
</tr>
<tr>
<td>7609</td>
<td>M</td>
<td>&quot;Moduli and Canonical Forms for Linear Dynamical Systems. II: The Topological Case&quot;, by M. Hazewinkel.</td>
<td></td>
</tr>
<tr>
<td>7611</td>
<td>M</td>
<td>&quot;On n-Isoclinic Groups&quot;, by M. Hazewinkel.</td>
<td></td>
</tr>
<tr>
<td>7612</td>
<td>S</td>
<td>&quot;Sample Extremes: An Elementary Introduction&quot;, by Laurens de Haan.</td>
<td></td>
</tr>
<tr>
<td>7614</td>
<td>E</td>
<td>&quot;Predictive Moments of Simultaneous Econometric Models, A Bayesian Approach&quot;, by H.K. van Dijk and T. Kloek.</td>
<td></td>
</tr>
<tr>
<td>7615</td>
<td>ES</td>
<td>&quot;Economics of Scale in Retailing&quot;, by B. Nooteboom.</td>
<td></td>
</tr>
<tr>
<td>7618</td>
<td>E</td>
<td>&quot;On Bayesian and Non-Bayesian Estimation of CES Production Functions within the Context of Two Alternative Models of Firm Behaviour&quot;, by S. Schim van der Loeff and R. Harkema.</td>
<td></td>
</tr>
<tr>
<td>7620</td>
<td>O</td>
<td>&quot;Projective Methods to Find a Feasible Solution to Systems of Linear Constraints&quot;, by J. Telgen and G. van der Hoek.</td>
<td></td>
</tr>
</tbody>
</table>